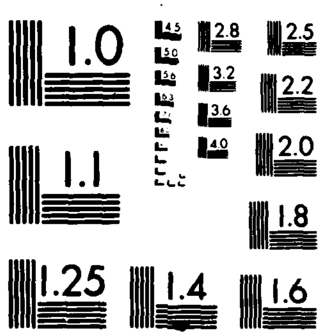
AD-A110 287   ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA   F/G 8/2
A DIGITAL TECHNIQUE FOR CONSTRUCTING VARIABLE-WIDTH LINES.(U)
MAR 82   R L ROSENTHAL
UNCLASSIFIED   ETL-R031   NL



MICROCOPY RESOLUTION TEST CHART

ADA110287

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL R031	2. GOVT ACCESSION NO. AD-A110287	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A DIGITAL TECHNIQUES FOR CONSTRUCTING VARIABLE-WIDTH LINES		5. TYPE OF REPORT & PERIOD COVERED Paper
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard L. Rosenthal		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Engineer Topographic Laboratories ATTN: ETL-LO Ft. Belvoir, VA 22060		12. REPORT DATE ~~14-~~20 March 1982
		13. NUMBER OF PAGES 9
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) LEVEL II		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC SELECTE FEB 1 1982 D S H

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

variable-width
laser platemaker
raster plotters
cartographic symbolization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper describes a digital technique for constructing variable-width cartographic lines. Specifically, an algorithm is developed for generating symbolized lines using multiply-stoked centerline data. Algorithmic development considers three factors: cartographic principles, inherent problems with existing algorithms, and hardware constraints imposed by specific output devices including the Defense Mapping Agency's Laser Platemaker. Software developed from the algorithm uses simple trigonometric functions and may be implemented in applications using vector data with vector or raster plotters.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

402192

A DIGITAL TECHNIQUE FOR CONSTRUCTING VARIABLE-WIDTH LINES

Richard L. Rosenthal
Cartographer
United States Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

ABSTRACT

This paper describes a digital technique for constructing variable-width cartographic lines. Specifically, an algorithm is developed for generating symbolized lines using multiply-stroked centerline data. Algorithmic development considers three factors: cartographic principles, inherent problems with existing algorithms, and hardware constraints imposed by specific output devices including the Defense Mapping Agency's Laser Platemaker. Software developed from the algorithm uses simple trigonometric functions and may be implemented in applications using vector data with vector or raster plotters.

TEXT

The ability to produce lines of different widths is a common requirement for automated map image generation. For vector equipment this capability is often found in hardware support such as pen selection on pen plotters or aperture selection on photoplotters. However, graphic devices using binary raster output do not support hardware 'pen' selection. On this equipment line widths must be generated by software simulation. In many cases the lines generated by software simulation do not maintain the standards of cartographic symbolization.

The U.S. Army Engineer Topographic Laboratory's (ETL) requirement for simulating line widths on binary raster devices relates to the development for the Defense Mapping Agency (DMA) of the Large-Format Laser Platemaker (LPM). Using the LPM as an output device for vector data involves the generation of lines in raster form using vector-to-raster conversion. Software used in the conversion process commonly creates a series of single pixels for each vector. The LPM is excessively fine when generating single pixel vectors; a line created from single pixels will have a width of only 1.5 mils. Such a line is not distinguishable.

A technique is needed for generating lines via vector-to-raster conversion which have a minimum distinguishable width associated with cartographic production (E.G. 4 to 5 mils). In addition to generating minimum widths, the technique should permit generating variable-width lines.

Software packages provided with binary raster devices (e.g. electro-static plotters) provide compatibility with basic pen plotter software. The vector-to-raster conversion is transparent to the applications programmer. These software packages often contain additional routines which take full advantage of the capabilities afforded by raster plotting including the generating of variable-width lines. Typically, the variable-width lines generated come from multiply stroked centerline data. The subsequent vector-to-raster conversion

takes the additional 'strokes' and generates a single pixel line for each. The algorithms which generate the variable-width lines repeatedly plot the centerline data parallel to itself at a distance equal to the step size between two adjacent pixels. Each stroke is the same length as the centerline and the resultant symbolized line has a rectangular appearance (Figure 1). Due to their proximity, the multiple single-pixel raster lines give the appearance of one multiple-pixel raster line. When the individual rectangular line segments are connected to form a line chain, the line has gaps at the vertices (Figure 2).

ALGORITHM DEVELOPMENT

An algorithm that generates lines that more closely resemble those produced by traditional drafting techniques follows. Lines, formed by drafting tools such as resevoir pens and scribers, have rounded ends. The developed multi-stroked algorithm generates variable-width lines which have rounded ends by varying the length of each stroke. A line chain formed by variable-width lines which have rounded ends has no gaps at the vertices (Figure 3).

Figure 1

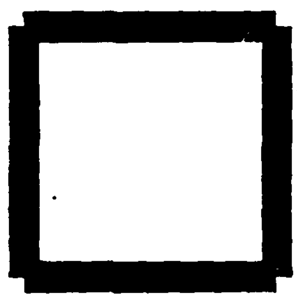


Figure 2

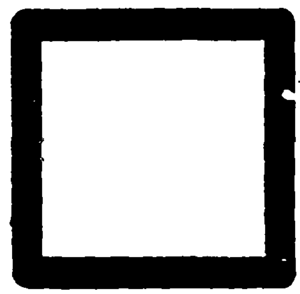


Figure 3

The lenght of each stroke is varied by calculating plot coordinates (Px1, Py1) and (Px2, Py2) which lie upon the circumference of semi-circles centered about the centerline end points (Figure 4). To calculate the plot end points (Px1, Py1) and (Px2, Py2) it is necessary to know the centerline end points (x1, y1) and (x2, y2), the width of a single stroke (W), and the number of strokes to generate (NTIM). Working at one end of the centerline, the plot coordinates (Px1, Py1) are found by

(1) $Px1 = x1 + dX$ and

(2) $Py1 = y1 + dY$

The coordinate off-sets (dX, dY) are calculated by

(3) $dX = R * \mathrm{COSINE}(c)$ and

(4) $dY = R * \mathrm{SINE}(c)$

where (R) is the radius of the 'thickening' semi-circle and (c) is the angle of the plot coordinates with respect ot the centerline end coordinates. The radius (R) is found by

(5) $R = 0.5 * NTIM * W$

where (NTIM) is the number of strokes and (W) is the stroke width.

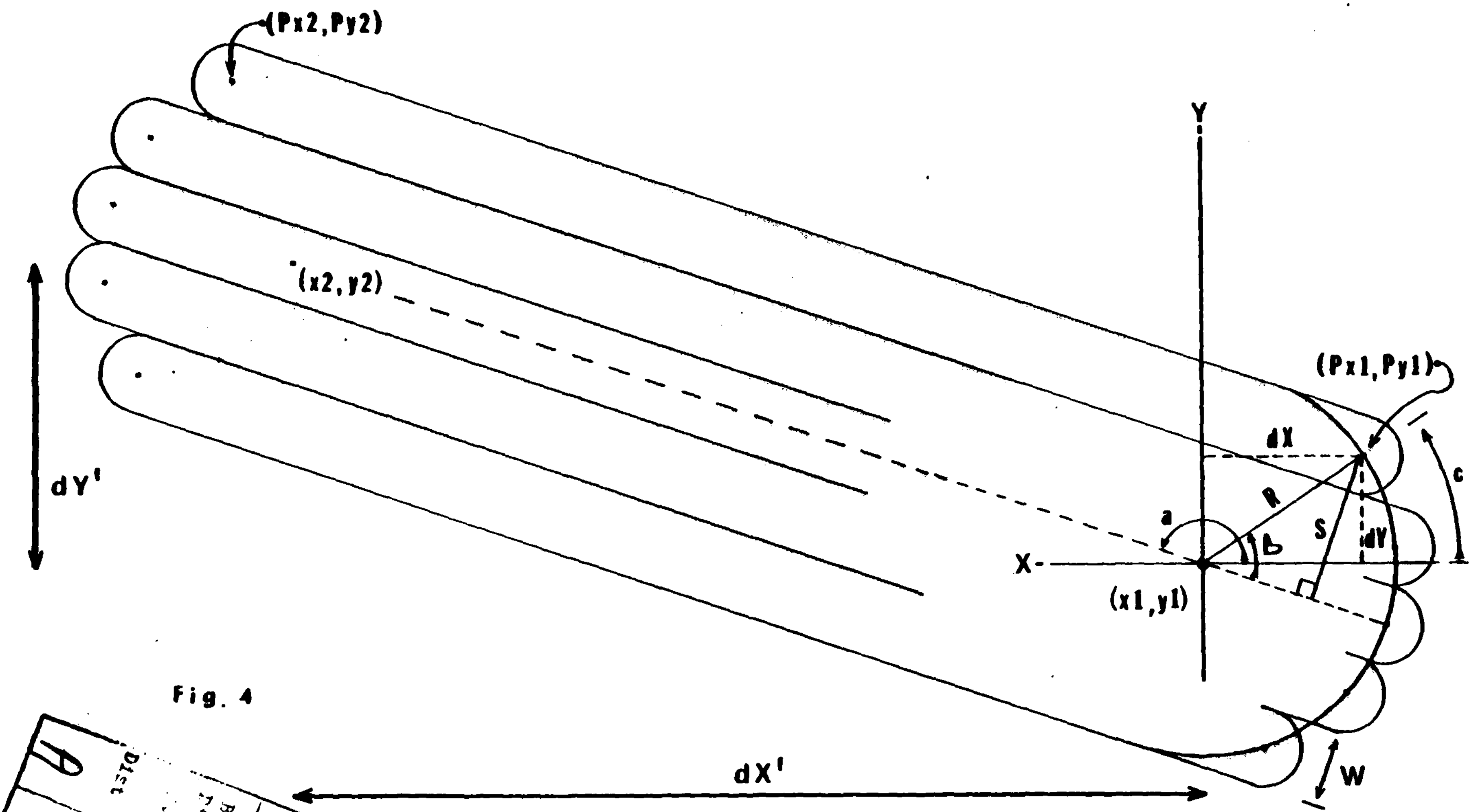


Fig. 4

The angle (c) is found by

(6) $c = (a - \pi) + b$

where (a) is the absolute angle of the centerline and (b) is the relative angle of the plot coordinates (Px1, Py1) measured from the centerline, extended. The angle (a) is found by

(7) $a = \mathrm{ARCTANGENT}(dY' / dX')$

where

(8) $dX' = x2 - x1$ and
(9) $dY' = y2 - y1$

The relative angle (b) is found by

(10) $b = \mathrm{ARCSINE}(S / R)$

where (S) is the distance of the generated plot coordinate from the centerline and (R) is the radius. The value of (S) or the length of the side opposite the relative angle (b) is calculated as

(11) $S = 0.5 * ((2 * I - NTIM - 1) * W)$

where (I) is the incremented stroke count, (NTIM) is the number of strokes, and (W) is the stroke width. It should be noted that the calculation of (S) in equation (11) compensates for the difference between odd and even stroke counts; the generated strokes always exhibit symmetry with respect to ehe vector centerline.

The above calculations take place at both ends of the centerline (x1, y1) and (x2, y2) and generate the plot coordinates (Px1, Py1) and (Px2, Py2) for each stroke. The total width of the multiply-stroked line equals the width (W) multiplied by the number of strokes generated (NTIM). The calculation can be changed such that the number of strokes (NTIM) equals the total width divided by the stroke width (W). This permits generating constant line widths regardless of available pen width. For applications using raster output devices, the stroke width (W) is set to the step size between pixels. On the Laser Platemaker this is approximately 0.001 inch. For applications using vector output devices, the stroke width (W) is set to the actual 'pen' width. This could be useful in insta-lations that have a limited selection of 'pen' widths.

The ability to produce lines of different widths is a common requirement for automated map image generation. On graphic devices where hardware line-width selection is unavailable, line-widths must be generated by software simulation. On high resolution devices such as the LPM a technique is needed for generating lines through software simulation which have a minimum width associated with cartographic symbolization standards. On devices where hardware line width selection is available but limited, a variable-width routine provides additional capability. This paper has described a digital technique for constructing variable-width lines based on multiply-stroked centerline data which addresses these requirements.

APPENDIX A

SUBROUTINE LISTING

```
C*********************************************************************
C-T     THICK
C-F     PLOT SUBROUTINE FOR VARIABLE-WIDTH LINES
C-L     FORTRAN IV-PLUS V02.51
C-S     RSX-11M VER 3.2 - OPERATING SYSTEM
C-A     RICHARD ROSENTHAL, USAETL, FT. BELVOIR, VIRGINIA 22060
C-D     1-DEC-81
C
C...    NOTE:  THIS ROUTINE IS DESIGNED TO BE CALLED IN THE SAME
C...           MANNER AS 'PLOT'.  IT MAY BE SUBSTITUTED FREELY FOR
C...           'PLOT' IN APPLICATIONS.  THE DEFAULT SETTINGS SIMPLY
C...           GENERATE A SINGLE STROKE BY DIRECTLY CALLING 'PLOT'.
C
C       CALL THICK(X, Y, IPEN)
C
C               X = X-COORDINATE (REAL)
C               Y = Y-COORDINATE (REAL)
C               IPEN = FUNCTION CODE (INTEGER)
C                    = +3 = MOVE TO (X, Y) - CALLS 'PLOT'
C                    = +2 = DRAW TO (X, Y)
C                    ALL OTHER VALUES CALL 'PLOT'
C
C       SUBROUTINES AND FUNCTIONS REFERENCED:
C       PLOT    $ASIN   $ATAN   $COS    $SIN
C
C       COMMON USED:  /THICK1/ NTIM, PNWD
C       COMMON MODIFIED:  /THICK1/ DFLTPN, NTIM, PNWD
C
        SUBROUTINE THICK(X, Y, IPEN)
C
        INTEGER I        !LOOP COUNTER
        INTEGER IFLAG    !FIRST CALL FLAG
        INTEGER INITIM   !INITIALIZATION STROKE COUNT
        INTEGER IP       !LOCAL PEN CODE
        INTEGER IPEN     !GLOBAL PEN CODE (PARAMETER)
        INTEGER NTIM     !LOCAL STROKE COUNT
        INTEGER PDOWN    !PEN DOWN
        INTEGER PUP      !PEN UP
C
        REAL AANG1       !ABSOLUTE ANGLE AT START COORDINATE
        REAL AANG2       !ABSOLUTE ANGLE AT END COORDINATE
        REAL DFLTPN      !DEFAULT PEN WIDTH
        REAL DX          !DELTA X
        REAL DY          !DELTA Y
        REAL INIPEN      !INITIALIZATION PEN WIDTH
        REAL LINANG      !LINE ANGLE, UNCORRECTED FOR QUADRANTS
        REAL LNANG1      !LINE ANGLE, CORRECTED, START TO END
        REAL LNANG2      !LINE ANGLE, CORRECTED, END TO START
        REAL PI          !3.14159...
        REAL PNWD        !LOCAL PEN WIDTH
        REAL PX1         !PLOTTING START ABSCISSA, GENERATED
        REAL PX2         !PLOTTING END ABSCISSA, GENERATED
        REAL PY1         !PLOTTING START ORDINATE, GENERATED
        REAL PY2         !PLOTTING END ORDINATE, GENERATED
        REAL RADIUS      !RADIUS OF THICKENING CIRCLE
        REAL ROTX1       !START ABSCISSA OFF-SET
```

```
      REAL ROTX2       !END ABSCISSA OFF-SET
      REAL ROTY1       !START ORDINATE OFF-SET
      REAL ROTY2       !END ORDINATE OFF-SET
      REAL SLOPE       !SLOPE OF LINE FROM START TO END
      REAL S1          !SIDE OPPOSITE ANGLE AT START COORDINATE
      REAL S2          !SIDE OPPOSITE ANGLE AT END COORDINATE
      REAL TANG1       !RELATIVE ANGLE AT START COORDINATE
      REAL TANG2       !RELATIVE ANGLE AT END COORDINATE
      REAL X           !GLOBAL END ABSCISSA (PARAMETER)
      REAL X1          !LOCAL START ABSCISSA
      REAL X2          !LOCAL END ABSCISSA
      REAL Y           !GLOBAL END ORDINATE (PARAMETER)
      REAL Y1          !LOCAL START ORDINATE
      REAL Y2          !LOCAL END ORDINATE
C
      COMMON /THICK1/ DFLTPN, NTIM, PNWD
C
C...  * VARIABLES MAY BE LOCALLY CONFIGURED
      DATA IFLAG  /0/          !FIRST CALL FLAG
      DATA INIPEN /0.005/      !INITIALIZATION PEN WIDTH *
      DATA INITIM /1/          !INITIALIZATION STROKE COUNT *
      DATA PDOWN  /2/          !PEN DOWN
      DATA PI     /3.14159/    !PIE
      DATA PUP    /3/          !PEN UP
      DATA X1     /0.0/        !LOCAL START ABSCISSA
      DATA Y1     /0.0/        !LOCAL START ORDINATE
C
C...  BEGIN
      IF (IFLAG .NE. 0) GOTO 90 !BRANCH AFTER FIRST CALL
C
C...  INITIALIZE COMMON VARIABLES
      DFLTPN = INIPEN !SET DEFAULT PEN WIDTH
      NTIM = INITIM !SET LOCAL STROKE COUNT
      PNWD = INIPEN !SET LOCAL PEN WIDTH
      IFLAG = 1 !SET FIRST CALL FLAG
C
90    CINTINUE
      X2 = X !LOCAL COPY
      Y2 = Y !LOCAL COPY
      IP = IPEN !LOCAL COPY
C
C...  SINGLE WIDTH LINE?
      IF (NTIM .EQ. 1) GOTO 190
C
C...  PEN DOWN MOVE?
      IF (IP .EQ. PDOWN) GOTO 100 !START ROUTINE
C
C...  PEN UP MOVE?
      IF (IP .EQ. PUP) GOTO 190
C
C...  UNDEFINED PEN CODE
      GOTO 190
C
100   CONTINUE
C...  PEN-DOWN MULTIPLY-STROKED LINE
      DX = X2 - X1 !DELTA X = END - START
      DY = Y2 - Y1
      IF ((DX .EQ. 0.0) .AND. (DY .EQ. 0.0)) GOTO 190 !NO MOVEMENT
      IF (DX .NE. 0.0) GOTO 110
C...  MANUAL CALCULATION WHEN DX = 0.0
```

```
          LINANG = -PI / 2 !FOLLOWS F4P $ATAN CONVENTION
          GOTO 115
C
110       CONTINUE
          SLOPE = DY / DX
          LINANG = ATAN(SLOPE) !LINANG = ARCTANGENT(SLOPE)
C
115       CONTINUE
C...      THE FOLLOWING CODE PLACES ANGLE IN PROPER QUADRANT
          IF ((DX .GT. 0.0) .AND. (DY .GE. 0.0)) GOTO 119 !QUAD 1
          IF ((DX .LE. 0.0) .AND. (DY .GT. 0.0)) GOTO 118 !QUAD 2
          IF ((DX .LT. 0.0) .AND. (DY .LE. 0.0)) GOTO 117 !QUAD 3
C
116       CONTINUE !CORRECT ANGLE FOR LINE GOING INTO QUAD 4
          LNANG1 = LINANG + (2 * PI)
          LNANG2 = LINANG + PI
          GOTO 120
C
117       CONTINUE !CORRECT ANGLE FOR LINE GOING INTO QUAD 3
          LNANG1 = LINANG + PI
          LNANG2 = LINANG
          GOTO 120
C
118       CONTINUE !CORRECT ANGLE FOR LINE GOING INTO QUAD 2
          LNANG1 = LINANG + PI
          LNANG2 = LINANG + (2 * PI)
          GOTO 120
C
119       CONTINUE !CORRECT ANGLE FOR LINE GOING INTO QUAD 1
          LNANG1 = LINANG
          LNANG2 = LINANG + PI
120       CONTINUE !LINE ANGLES ARE NOW CORRECTED
C
C...      ONE OF TWO RADIUS CALCULATIONS ARE POSSIBLE:
          RADIUS = 0.5 * NTIM * PNWD
C-O       RADIUS = 0.5 * (NTIM - 1) * PNWD
C
          DO 130 I = 1, NTIM !ONE LOOP FOR EACH PEN STROKE
C...        CALCULATE LENGTH OF SIDE OPPOSITE RELATIVE ANGLE
            S1 = 0.5 * ((2 * I - NTIM - 1) * PNWD) !AT START POINT
            S2 = -S1 !AT END POINT
C
C...        CALCULATE RELATIVE ANGLE
            TANG1 = ASIN(S1 / RADIUS) !AT START POINT
            TANG2 = ASIN(S2 / RADIUS) !AT END POINT
C
C...        CALCULATE ABSOLUTE ANGLE
            AANG1 = (LNANG1 - PI) + TANG1 !AT START POINT
            AANG2 = (LNANG2 - PI) + TANG2 !AT END POINT
C
C...        CALCULATE START COORDINATE OFF-SET
            ROTX1 = RADIUS * COS(AANG1)
            ROTY1 = RADIUS * SIN(AANG1)
C
C...        CALCULATE END COORDINATE OFF-SET
            ROTX2 = RADIUS * COS(AANG2)
            ROTY2 = RADIUS * SIN(AANG2)
C
C...        CALCULATE PLOTTING START COORDINATE
            PX1 = X1 + ROTX1 !PLOTTING = START + OFF-SET
```

```
            PY1 = Y1 + ROTY1
C
C...        CALCULATE PLOTTING END COORDINATE
            PX2 = X2 + ROTX2 !PLOTTING = END + OFF-SET
            PY2 = Y2 + ROTY2
C
C...        DRAW FROM GENERATED START TO GENERATED END COORDINATE
            CALL PLOT(PX1, PY1, PUP) !GRAPHIC MOVE
            CALL PLOT(PX2, PY2, PDOWN) !GRAPHIC DRAW
130         CONTINUE !DO ANOTHER STROKE
C
        IP = PUP !PEN UP
C
190     CONTINUE
C...    UPDATE LOCAL START COORDINATE
        X1 = X2 !NEW START = OLD END POINT
        Y1 = Y2
C
C...    UPDATE DEVICE PLOT ROUTINE
        CALL PLOT(X2, Y2, IP)
C
        RETURN
        END !OF THICK SUBROUTINE
C***********************************************************************
C-T     STROKE
C-F     PLOT SUBROUTINE TO SET NUMBER OF PEN STROKES
C-L     FORTRAN IV-PLUS V02.51
C-S     RSX-11M VER 3.2 - OPERATING SYSTEM
C-A     RICHARD ROSENTHAL, USAETL, FT. BELVOIR, FIVGINIA 22060
C-D     1-DEC-81
C
C...    NOTE:  THIS ROUTINE IS REQUIRED TO CHANGE THE NUMBER OF
C...           STROKES (NTIM) USED IN 'THICK'
C
C       CALL STROKE(NTIMES)
C
C              NTIMES = NUMBER OF PEN STROKES PER MULTIPLY-
C                       STROKED LINE (INTEGER)
C
C       COMMON MODIFIED:  /THICK1/ NTIM
C
        SUBROUTINE STROKE(NTIMES)
C
        INTEGER NTIM      !LOCAL STROKE COUNT
        INTEGER NTIMES    !GLOBAL STROKE COUNT (PARAMETER)
C
        REAL DFLTPN       !DEFAULT PEN WIDTH
        REAL PNWD         !LOCAL PEN WIDTH
C
        COMMON /THICK1/ DFLTPN, NTIM, PNWD
C
C...    BEGIN
        NTIM = NTIMES !LOCAL COPY
        IF(NTIM .LT. 1) NTIM = 1 !SAFTY CHECK
        RETURN
        END !OF STROKE SUBROUTINE
C***********************************************************************
C-T     PEN
C-F     PLOT SUBROUTINE TO SET WIDTH OF PLOTTER PEN
C-L     FORTRAN IV-PLUS V02.51
```

```
C-S     RSX-11M VER 3.2 - OPERATING SYSTEM
C-A     RICHARD ROSENTHAL, USAETL, FT. BELVOIR, VIRGINIA 22060
C-D     1-DEC-81
C
C...    NOTE:  THIS ROUTINE IS REQUIRED TO CHANGE THE 'PEN' WIDTH
C...           (PNWD) USED IN 'THICK'
C
C       CALL PEN(PENWID)
C
C               PENWID = WIDTH OF PLOTTER PEN (REAL)
C
C       COMMON USED:  /THICK1/ DELTPN
C       COMMON MODIFIED:  /THICK1/ PNWD
C
        SUBROUTINE PEN(PENWID)
C
        INTEGER NTIM     !LOCAL STROKE COUNT
C
        REAL DFLTPN      !DEFAULT PEN WIDTH
        REAL PENWID      !GLOBAL PEN WIDTH )PARAMETER)
        REAL PNWD        !LOCAL PEN WIDTH
C
        COMMON /THICK1/ DFLTPN, NTIM, PNWD
C
C...    BEGIN
        PNWD = PENWID !LOCAL COPY
        IF(PNWD .LE. 0.0) PNWD = DFLTPN !SAFTY CHECK
        RETURN
        END !OF PEN SUBROUTINE
C
C...    THANKS TO DOUG CALDWELL FOR HIS EDITORIAL ASSISTANCE*  .
```